EAST Search History

Ref #	Hits	Search Query	DBs	Default Operator	Plurals	Time Stamp
S1	1546	IPv4 or IPv6	USPAT; USOCR; EPO; JPO; IBM_TDB	OR	OFF	2006/06/13 11:52
S2	349	S1 and (rout$3 and prefix$3)	USPAT; USOCR; EPO; JPO; IBM_TDB	OR	OFF	2006/06/13 11:54
S3	293	S2 and (table$3)	USPAT; USOCR; EPO; JPO; IBM_TDB	OR	OFF	2006/06/13 11:55
S4	281	S3 and @ad<="20031124"	USPAT; USOCR; EPO; JPO; IBM_TDB	OR	OFF	2006/06/13 16:32
S5	46	S4 and ((protocol$3 with data$2 with unit$2) or PDU$3)	USPAT; USOCR; EPO; JPO; IBM_TDB	OR	OFF	2006/06/13 13:13
S7	22	S5 and (window$3)	USPAT; USOCR; EPO; JPO; IBM_TDB	OR	OFF	2006/06/13 12:54
S8	11	S5 and (slid$4 same window$3)	USPAT; USOCR; EPO; JPO; IBM_TDB	OR	OFF	2006/06/13 12:54
S9	51	("5278789" | "5390173" | "5414704" | "5423015" | "5459717" | "5473607" | "5499295" | "5524254" | "5555398" | "5568477" | "5579301" | "5644784" | "5652579" | "5696899" | "5742613" | "5748631" | "5781549" | "5787084" | "5790539" | "5802052" | "5802287" | "5825772" | "5828653" | "5831980" | "5842038" | "5845081" | "5887187" | "5892922" | "5898687" | "5909686" | "5918074" | "5940596" | "5950205" | "5987507" | "6011795" | "6041053" | "6052683" | "6061351" | "6067574" | "6119196" | "6175902" | "6185185" | "6223172" | "6430527" | "6553002" | "6581106" | "6594268" | "6631419" | "6658482").PN. OR ("6826561").URPN.	US-PGPUB; USPAT; USOCR	OR	OFF	2006/06/13 13:09

EAST Search History

S10	10243	((protocol$3 with data$2 with unit$2) or PDU$3)	USPAT; USOCR; EPO; JPO; IBM_TDB	OR	OFF	2006/06/13 13:28
S11	1302	S10 and (rout$3 same table$3)	USPAT; USOCR; EPO; JPO; IBM_TDB	OR	OFF	2006/06/13 13:30
S12	103	S11 and (window$3 and prefix$3 and (byt$4 or bit$3))	USPAT; USOCR; EPO; JPO; IBM_TDB	OR	OFF	2006/06/13 13:34
S13	101	S12 and @ad<="20031124"	USPAT; USOCR; EPO; JPO; IBM_TDB	OR	OFF	2006/06/13 13:54
S14	51	"6018524"	USPAT; USOCR; EPO; JPO; IBM_TDB	OR	OFF	2006/06/13 13:55
S16	1	"6018524".pn.	USPAT; USOCR; EPO; JPO; IBM_TDB	OR	OFF	2006/06/13 13:55
S17	2082	((hash$3) or (hash$3 with table)) and rout$3 and table$2 and prefix$4	US-PGPUB; USPAT; USOCR; EPO; JPO; DERWENT; IBM_TDB	OR	OFF	2006/06/13 16:31
S18	845	S17 and @ad<="20031124"	USPAT; USOCR; EPO; JPO; IBM_TDB	OR	OFF	2006/06/13 16:32
S22	308	S18 and ((protocol$3 or IP$2) and offset$3)	USPAT; USOCR; EPO; JPO; IBM_TDB	OR	OFF	2006/06/13 17:35
S23	195	S22 and (window$3)	USPAT; USOCR; EPO; JPO; IBM_TDB	OR	OFF	2006/06/13 17:53
S24	26	S23 and (prefix$3 same match$4) and (bit$2 or byt$3)	USPAT; USOCR; EPO; JPO; IBM_TDB	OR	OFF	2006/06/13 17:55

S25	2082	((hash$3) or (hash$3 with table)) and rout$3 and table$2 and prefix$4	US-PGPUB; USPAT; USOCR; EPO; JPO; DERWENT; IBM_TDB	OR	OFF	2006/06/14 09:59
S26	845	S25 and @ad<="20031124"	USPAT; USOCR; EPO; JPO; IBM_TDB	OR	OFF	2006/06/14 09:59
S27	308	S26 and ((protocol$3 or IP$2) and offset$3)	USPAT; USOCR; EPO; JPO; IBM_TDB	OR	OFF	2006/06/14 09:59
S28	195	S27 and (window$3)	USPAT; USOCR; EPO; JPO; IBM_TDB	OR	OFF	2006/06/14 09:59
S29	26	S28 and (prefix$3 same match$4) and (bit$2 or byt$3)	USPAT; USOCR; EPO; JPO; IBM_TDB	OR	OFF	2006/06/14 10:02
S30	1	S29 and (window$2 same offset$2)	USPAT; USOCR; EPO; JPO; IBM_TDB	OR	OFF	2006/06/14 10:04
S31	12	S27 and (window$2 same offset$2)	USPAT; USOCR; EPO; JPO; IBM_TDB	OR	OFF	2006/06/14 10:05
S32	51	("5278789" | "5390173" | "5414704" | "5423015" | "5459717" | "5473607" | "5499295" | "5524254" | "5555398" | "5568477" | "5579301" | "5644784" | "5652579" | "5696899" | "5742613" | "5748631" | "5781549" | "5787084" | "5790539" | "5802052" | "5802287" | "5825772" | "5828653" | "5831980" | "5842038" | "5845081" | "5887187" | "5892922" | "5898687" | "5909686" | "5918074" | "5940596" | "5950205" | "5987507" | "6011795" | "6041053" | "6052683" | "6061351" | "6067574" | "6119196" | "6175902" | "6185185" | "6223172" | "6430527" | "6553002" | "6581106" | "6594268" | "6631419" | "6658482").PN. OR ("6826561"). URPN.	US-PGPUB; USPAT; USOCR	OR	OFF	2006/06/14 11:27

S33	48	("6052683" "6934252" "6658482" "6141738" "6243720" "6212184" "6212184" "6675163" "6876655" "6581106" "6956858" "6633865" "6947931" "6067574" "6181698" "6192051" "6307855" "7031320" "6963924" "6678678" "5983223" "6434144" "6011795" "6993031" "6018524" "5946679" "6665297" "6826561" "6223172" "6223172" "6782382" "6985483" "6147976" "6449256" "6549536" "6975631" "6691171" "6289013" "6341130" "6792423" "6201755" "6571313" "6295296" "6563823" "6526055" "6631419" "6563956" "6876774" "6539369" "6691218").pn.	US-PGPUB; USPAT; USOCR	OR	OFF	2006/06/14 11:28
S34	1	(US-5802287-$).did.	USPAT	OR	OFF	2006/06/14 14:29
S36	2091	((hash$3) or (hash$3 with table)) and rout$3 and table$2 and prefix$4	US-PGPUB; USPAT; USOCR; EPO; JPO; DERWENT; IBM_TDB	OR	OFF	2006/06/15 14:55
S37	845	S36 and @ad<="20031124"	USPAT; USOCR; EPO; JPO; IBM_TDB	OR	OFF	2006/06/15 14:55
S38	308	S37 and ((protocol$3 or IP$2) and offset$3)	USPAT; USOCR; EPO; JPO; IBM_TDB	OR	OFF	2006/06/15 14:57
S39	35	S38 and (window$2 with size$1)	USPAT; USOCR; EPO; JPO; IBM_TDB	OR	OFF	2006/06/15 15:10
S40	207	S38 and ((window$2 with size$1) or filter$3)	USPAT; USOCR; EPO; JPO; IBM_TDB	OR	OFF	2006/06/15 15:39
S41	41	S38 and ((window$2 with size$1) or ((filter$3 or window$2) same slid$4))	USPAT; USOCR; EPO; JPO; IBM_TDB	OR	OFF	2006/06/15 15:40
S43	1	"20040085953"	US-PGPUB	OR	OFF	2006/07/03 16:38
S44	1	"20060050690"	US-PGPUB	OR	OFF	2006/07/03 16:38
S45	3	("6721316"|"6611832"|"6563823"). pn.	USPAT	OR	OFF	2006/07/03 16:47
S46	1	(US-20040085953-$).did.	US-PGPUB	OR	OFF	2006/07/06 13:26

S47	1	S46 and (stor$3 same tabl$3)	US-PGPUB; USPAT	OR	OFF	2006/07/06 13:27

Google Scholar BETA

[Search] Advanced Scholar Search
Scholar Preferences
Scholar Help

Scholar Results **1** - **10** of about **1,240** for **Binary search, prefix length, window, offset**. (**0.13** seconds)

All articles Recent articles

On the implementation of minimum redundancy **prefix** codes - group of 6 »
A Moffat, A Turpin - Communications, IEEE Transactions on, 1997 - ieeexplore.ieee.org
... the total number (over all symbols encoded) of codewords emitted of **length** for A ...
redundancy alphabetic **prefix** code (or optimal **binary search** tree) can then ...
Cited by 26 - Web Search - BL Direct

Better OPM/L Text Compression - group of 3 »
T Bell - Communications, IEEE Transactions on [legacy, pre-1988], 1986 - ieeexplore.ieee.org
... Section V shows that a **binary search** tree data structure can be used to ... the strings,
M(s, t), is defined to be the **length** of the longest common **prefix** of s ...
Cited by 19 - Web Search

B-Tree Indexes and CPU Caches - group of 7 »
G Graefe, P Larson - Proceedings of the International Conference on Data ..., 2001 - doi.ieeecomputersociety.org
... Software**Windows** 9 2 6 Software**Windows**\\Explorer 16 ... would store the key "\\Office",
the **prefix length** would be ... page, yet help in a **binary search** because the ...
Cited by 19 - Web Search - BL Direct

Routing interval: a new concept for IP lookups
PC Wang, CT Chan, SC Hu, YC Chen - ATM (ICATM 2001) and High Speed Intelligent Internet ..., 2001 - ieeexplore.ieee.org
... The number of prefixes whose **length** ... a pointer to either the next node or the **prefix**
table. ... also address how to improve the performance of **binary search** on hash ...
Web Search

ON THE IPLEWXVTATIN 0 MINIMUM-REDUNDANCY **PREFIX** CODES - group of 2 »
A Moffatt, A Turpini - doi.ieeecomputersociety.org
... In fact the distribution of **length**'s in ... for the array can substantially improve upon
a **binary search**. ... devise a minimum-redundancy alphabetic **prefix** code for ...
Web Search

DIFFERENTIAL COMPRESSION: A GENERALIZED SOLUTION FOR **BINARY** FILES - group of 4 »
RC Burns - 1996 - almaden.ibm.com
... The **binary** algorithms under consideration operate on data ... that was not in the original
string **prefix**, ... be the longest without performing exhaustive **search** for ...
Cited by 9 - View as HTML - Web Search - Library Search

IP lookups using multiway and multicolumn **search** - group of 14 »
B Lampson, V Srinivasan, G Varghese - IEEE/ACM Transactions on Networking (TON), 1999 - portal.acm.org
... Additionally, if there are prefixes of longer **length** with that **prefix** the array
element stores a pointer to a **binary search** table/tree that contains ...
Cited by 199 - Web Search - BL Direct

A general practical approach to pattern matching over Ziv-Lempel compressed text - group of 8 »
G Navarro, M Raffinot - Proc. 10th Ann. Symp. on Combinatorial Pattern Matching - Springer
... If we consider m ≤ w, then we keep the state of the **search** in a computer word D,
whose i-th bit tells whether the **prefix** of **length** i of the pattern ...
Cited by 51 - Web Search - BL Direct

The s2-tree: An index structure for subsequence matching of spatial objects - group of 3 »
H Wang, CS Perng - Proceedings of the 5th Pacific-Asic Conference on Knowledge ..., 2003 - Springer
... Output: symbol set is a set of **binary** characters in ... one subsequence on the edges
matches a **prefix** of L ... 5. **Search**() performs subsequence matching on a suffix tree ...
Cited by 6 - Web Search - BL Direct

The methods of improving the compression ratio of LZ77 family datacompression algorithms

Terms used
Forwarding engine for fast routing lookups and **updates**

Found **22,650** of **178,880**

Sort results by: relevance
Display results: expanded form

Save results to a Binder
Search Tips
□ Open results in a new window

Try an Advanced Search
Try this search in The ACM Guide

Results 1 - 20 of 200 Result page: **1** 2 3 4 5 6 7 8 9 10 next
Best 200 shown

Relevance scale □ ▭ ▬ ■ ■

1 Fast incremental updates for pipelined forwarding engines ■
Anindya Basu, Girija Narlikar
June 2005 **IEEE/ACM Transactions on Networking (TON)**, Volume 13 Issue 3
Publisher: ACM Press
Full text available: pdf(941.54 KB) Additional Information: full citation, abstract, references, index terms

Pipelined ASIC architectures are increasingly being used in forwarding engines for high-speed IP routers. We explore optimization issues in the design of memory-efficient data structures that support fast incremental updates in such forwarding engines. Our solution aims to balance the memory utilization across the multiple pipeline stages. We also propose a series of optimizations that minimize the disruption to the forwarding process caused by route updates. These optimizations reduce the updat ...

Keywords: core routers, packet forwarding, pipelined IP lookup, route updates

2 Full papers: Tree bitmap: hardware/software IP lookups with incremental updates ■
Will Eatherton, George Varghese, Zubin Dittia
April 2004 **ACM SIGCOMM Computer Communication Review**, Volume 34 Issue 2
Publisher: ACM Press
Full text available: pdf(189.39 KB) Additional Information: full citation, abstract, references

Even with the significant focus on IP address lookup in the published literature as well as focus on this market by commercial semiconductor vendors, there is still a challenge for router architects to find solutions that simultaneously meet 3 criteria: scaling in terms of lookup speeds as well as table sizes, the ability to perform high speed updates, and the ability to fit into the overall memory architecture of an Level 3 forwarding engine or packet processor with low systems cost overhead. I ...

3 Small forwarding tables for fast routing lookups ■
Mikael Degermark, Andrej Brodnik, Svante Carlsson, Stephen Pink
October 1997 **ACM SIGCOMM Computer Communication Review , Proceedings of the ACM SIGCOMM '97 conference on Applications, technologies, architectures, and protocols for computer communication SIGCOMM '97**, Volume 27 Issue 4
Publisher: ACM Press
Full text available: pdf(1.62 MB) Additional Information: full citation, abstract, references, citings, index terms

For some time, the networking community has assumed that it is impossible to do IP routing lookups in software fast enough to support gigabit speeds. IP routing lookups must find the routing entry with the *longest matching prefix*, a task that has been thought to require hardware support at lookup frequencies of millions per second.We present a forwarding table data structure designed for quick routing lookups. Forwarding tables are

Keywords: data communications, internetworking, packet switching, routing

8 Lookups: Dynamic pipelining: making IP-lookup truly scalable

Jahangir Hasan, T. N. Vijaykumar

August 2005 **Proceedings of the 2005 conference on Applications, technologies, architectures, and protocols for computer communications SIGCOMM '05**

Publisher: ACM Press

Full text available: pdf(194.92 KB) Additional Information: full citation, abstract, references, index terms

A truly scalable IP-lookup scheme must address five challenges of scalability, namely: routing-table size, lookup throughput, implementation cost, power dissipation, and routing-table update cost. Though several IP-lookup schemes have been proposed in the past, none of them do well in all the five scalability requirements. Previous schemes pipeline tries by mapping trie levels to pipeline stages. We make the fundamental observation that because this mapping is static and oblivious of the prefix ...

Keywords: IP-lookup, longest prefix matching, pipelined, scalable, tries

9 Scalable high-speed prefix matching

Marcel Waldvogel, George Varghese, Jon Turner, Bernhard Plattner

November 2001 **ACM Transactions on Computer Systems (TOCS)**, Volume 19 Issue 4

Publisher: ACM Press

Full text available: pdf(933.02 KB) Additional Information: full citation, abstract, references, citings, index terms

Finding the longest matching prefix from a database of keywords is an old problem with a number of applications, ranging from dictionary searches to advanced memory management to computational geometry. But perhaps today's most frequent best matching prefix lookups occur in the Internet, when forwarding packets from router to router. Internet traffic volume and link speeds are rapidly increasing; at the same time, a growing user population is increasing the size of routing tables against which p ...

Keywords: collision resolution, forwarding lookups, high-speed networking

10 Faster IP lookups using controlled prefix expansion

V. Srinivasan, George Varghese

June 1998 **ACM SIGMETRICS Performance Evaluation Review , Proceedings of the 1998 ACM SIGMETRICS joint international conference on Measurement and modeling of computer systems SIGMETRICS '98/PERFORMANCE '98**, Volume 26 Issue 1

Publisher: ACM Press

Full text available: pdf(1.31 MB) Additional Information: full citation, abstract, references, citings, index terms

Internet (IP) address lookup is a major bottleneck in high performance routers. IP address lookup is challenging because it requires *a longest matching prefix* lookup. It is compounded by increasing routing table sizes, increased traffic, higher speed links, and the migration to 128 bit IPv6 addresses. We describe how IP lookups can be made faster using a new technique called *controlled prefix expansion*. Controlled prefix expansion, together with optimization techniques based on dyn ...

11 Scalable high speed IP routing lookups

Marcel Waldvogel, George Varghese, Jon Turner, Bernhard Plattner

October 1997 **ACM SIGCOMM Computer Communication Review , Proceedings of the ACM SIGCOMM '97 conference on Applications, technologies, architectures, and protocols for computer communication SIGCOMM '97**, Volume 27 Issue 4

Publisher: ACM Press

Full text available: pdf(1.66 MB) Additional Information: full citation, abstract, references, citings, index terms

Internet address lookup is a challenging problem because of increasing routing table sizes, increased traffic, higher speed links, and the migration to 128 bit IPv6 addresses. IP routing lookup requires computing the best matching prefix, for which standard solutions like hashing were believed to be inapplicable. The best existing solution we know of, BSD radix tries, scales badly as IP moves to 128 bit addresses. Our paper describes a new algorithm for best matching prefix using binary search o ...

12 Lookups: Fast hash table lookup using extended bloom filter: an aid to network processing
Haoyu Song, Sarang Dharmapurikar, Jonathan Turner, John Lockwood
August 2005 **Proceedings of the 2005 conference on Applications, technologies, architectures, and protocols for computer communications SIGCOMM '05**
Publisher: ACM Press
Full text available: pdf(338.54 KB) Additional Information: full citation, abstract, references, index terms

Hash tables are fundamental components of several network processing algorithms and applications, including route lookup, packet classification, per-flow state management and network monitoring. These applications, which typically occur in the data-path of high-speed routers, must process and forward packets with little or no buffer, making it important to maintain wire-speed throughout. A poorly designed hash table can critically affect the worst-case throughput of an application, since the num ...

Keywords: forwarding, hash table

13 Measurement: The impact of address allocation and routing on the structure and implementation of routing tables
Harsha Narayan, Ramesh Govindan, George Varghese
August 2003 **Proceedings of the 2003 conference on Applications, technologies, architectures, and protocols for computer communications**
Publisher: ACM Press
Full text available: pdf(148.92 KB) Additional Information: full citation, abstract, references, index terms

The recent growth in the size of the routing table has led to an interest in quantitatively understanding both the causes (*eg* multihoming) as well as the effects (*eg* impact on router lookup implementations) of such routing table growth. In this paper, we describe a new model called **ARAM** that defines the structure of routing tables of any given size. Unlike simpler empirical models that work backwards from effects (*eg* current prefix length distributions), **ARAM** a ...

Keywords: IP lookups, modeling, routing tables

14 A Tree Based Router Search Engine Architecture with Single Port Memories
Florin Baboescu, Dean M. Tullsen, Grigore Rosu, Sumeet Singh
May 2005 **ACM SIGARCH Computer Architecture News , Proceedings of the 32nd Annual International Symposium on Computer Architecture ISCA '05,** Volume 33 Issue 2
Publisher: IEEE Computer Society, ACM Press
Full text available: pdf(293.29 KB) Additional Information: full citation, abstract, index terms

Pipelined forwarding engines are used in core routers to meet speed demands. Tree-based searches are pipelined across a number of stages to achieve high throughput, but this results in unevenly distributed memory. To address this imbalance, conventional approaches use either complex dynamic memory allocation schemes or over-provision each of the pipeline stages. This paper describes the microarchitecture of a novel network search processor which provides both high execution throughput and balanc ...

15 Multicore software: High-performance IPv6 forwarding algorithm for multi-core and multithreaded network processor
Xianghui Hu, Xinan Tang, Bei Hua
March 2006 **Proceedings of the eleventh ACM SIGPLAN symposium on Principles and practice of parallel programming PPoPP '06**